

37장. Set과 Map

37.1 Set

1. Set 객체는 중복되지 않는 유일한 값들의 집합(set)이다.

Set 객체는 배열과 유사하지만 다음과 같은 차이가 있다.

구분	배열	Set 객체
동일한 값을 중복하여 포함할 수 있다.	O	X
요소 순서에 의미가 있다.	O	X
인덱스로 요소에 접근할 수 있다.	O	X

2. 이러한 Set 객체 특성은 수학적 집합의 특성과 일치.

Set은 수학적 집합을 구현하기 위한 자료구조.

37.1.1 Set 객체의 생성

1. Set 객체는 Set 생성자 함수로 생성한다. 인수를 전달하지 않으면 빈 객체 생성.

2. Set 생성자 함수는 이터러블을 인수로 전달받아 Set 객체를 생성한다.

이때 이터러블의 중복된 값은 Set 객체에 요소로 저장되지 X.

➡ 배열에서 중복된 요소를 제거할 수 있다.

37.1.2 요소 개수 확인

1. Set 객체 요소 개수를 확인할 때는 Set.prototype.size 프로퍼티 사용
2. size 프로퍼티는 setter 함수 없이, getter 함수만 존재하는 접근자 프로퍼티.
따라서, Set 객체의 요소 개수 변경 불가능

37.1.3 요소 추가

1. Set 객체 요소 추가할 때는 Set.prototype.add 메서드 사용
2. add 메서드를 호출한 후에 연속적으로 add 메서드를 호출할 수 있다.

3. 중복된 요소의 추가는 허용 X

4. Set 객체는 객체, 배열과 같이 자바스크립트 모든 값을 요소로 저장할 수 있다.

37.1.4 요소 존재 여부 확인

1. 특정 요소가 존재하는지 확인하려면 Set.prototype.has 메서드 사용

→ 존재 여부를 나타내는 불리언 값 반환

37.1.5 요소 삭제

1. 특정 요소를 삭제하려면 Set.prototype.delete 메서드 사용

2. delete 메서드에는 인덱스가 아니라, 삭제하려는 요소값을 인수로 전달해야 함.

Set 객체에는 순서에 의미 X (=인덱스를 갖지 않음).

37.1.6 요소 일괄 삭제

1. Set 객체 모든 요소를 일괄 삭제하려면 Set.prototype.clear 메서드 사용

언제나 undefined 반환

37.1.7 요소 순회

1. 객체의 요소를 순회하려면 Set.prototype.forEach 메서드 사용.

Array.prototype.forEach 메서드와 유사하게 콜백 함수와 forEach 메서드 콜백 함수 내부에서 this로 사용될 객체를 인수로 전달.

- 첫 번째 인수 : 현재 순회 중인 요소값
- 두 번째 인수 : 현재 순회 중인 요소값
- 세 번째 인수 : 현재 순회 중인 Set 객체 자체

2. **Set 객체는 이터러블이다.** 따라서 for ... of 문으로 순회할 수 있으며,

스프레드 문법, 배열 디스트럭처링의 대상이 될 수도 있다.

37.1.8 집합 연산

교집합

: 교집합 $A \cap B$ 는 집합 A, B의 공통 요소로 구성.

합집합

: 합집합 $A \cup B$ 는 집합 A, B의 중복 없는 모든 요소로 구성.

차집합

: 차집합 $A - B$ 는 A에는 존재하지만, B에는 존재하지 않는 요소로 구성.

부분 집합과 상위 집합

집합 A가 B에 포함되는 경우 ($A \subseteq B$), A는 B의 부분 집합(subset), B는 A의 상위 집합(superset).

37.2 Map

1. Map 객체는 키와 값의 쌍으로 이루어진 컬렉션.

객체와 유사하지만 다음과 같은 차이가 있다.

구분	객체	Map 객체
키로 사용할 수 있는 값	문자열 or 심벌 값	객체를 포함한 모든 값
이터러블	X	O
요소 개수 확인	<code>Object.keys(obj).length</code>	<code>map.size</code>

37.2.1 Map 객체의 생성

1. Map 객체는 Map 생성자 함수로 생성한다. 인수를 전달하지 않으면 빈 객체 생성.
2. Map 생성자 함수는 이터러블을 인수로 전달받아 Map 객체를 생성한다.
이때 인수로 전달되는 이터러블은 키와 값의 쌍으로 이루어진 요소로 구성돼야 함.
3. 중복된 키를 갖는 요소가 존재하면 값이 덮어써짐 → 따라서 중복된 키 요소 존재 불가능

```
const map = new Map([['key1', 'value1'], ['key1', 'value2']]);  
console.log(map); // Map(1) {"key1" => "value2"}
```

37.2.2 요소 개수 확인

1. 요소 개수를 확인할 때는 `Map.prototype.size` 프로퍼티를 사용
2. `size` 프로퍼티는 `setter` 함수 없이, `getter` 함수만 존재하는 접근자 프로퍼티.
따라서, `Map` 객체의 요소 개수 변경 불가능

37.2.3 요소 추가

1. 요소를 추가할 때는 `Map.prototype.set` 메서드를 사용
2. `set` 메서드를 호출한 후에 연속적으로 `set` 메서드를 호출할 수 있다.
3. 중복된 요소의 추가는 허용 X
4. `Map` 객체는 키 타입에 제한 X. 모든 값을 키로 사용할 수 있다.

37.2.4 요소 취득

1. 특정 요소를 취득하려면 `Map.prototype.get` 메서드 사용.
인수로 전달한 키를 갖는 요소가 존재하지 않으면 `undefined` 반환

37.2.5 요소 존재 여부 확인

1. 특정 요소가 존재하는지 확인하려면 `Map.prototype.has` 메서드 사용
→ 존재 여부를 나타내는 불리언 값 반환

37.2.6 요소 삭제

1. 특정 요소를 삭제하려면 `Map.prototype.delete` 메서드 사용
2. `delete` 메서드는 삭제 성공 여부를 나타내는 불리언 값을 반환.
따라서, `set` 메서드와 달리 연속적으로 호출(`method chainig`)할 수 없다.

37.2.7 요소 일괄 삭제

1. Set 객체 모든 요소를 일괄 삭제하려면 Set.prototype.clear 메서드 사용
언제나 undefined 반환

37.2.8 요소 순회

1. 객체의 요소를 순회하려면 Map.prototype.forEach 메서드 사용.
Array.prototype.forEach 메서드와 유사하게 콜백 함수와 forEach 메서드 콜백 함수 내부에서 this로 사용될 객체를 인수로 전달.
 - 첫 번째 인수 : 현재 순회 중인 요소값
 - 두 번째 인수 : 현재 순회 중인 요소키
 - 세 번째 인수 : 현재 순회 중인 Map 객체 자체
2. **Map 객체는 이터러블이다.** 따라서 for ... of 문으로 순회할 수 있으며,
스프레드 문법, 배열 디스트럭처링의 대상이 될 수도 있다.
3. Map 객체는 이터러블이면서 동시에 이터레이터인 객체를 반환하는 메서드 제공
 - Map.prototype.keys : 요소키를 값으로 갖는 이터러블이면서 동시에 이터레이터인 객체를 반환
 - Map.prototype.values : 요소값을 값으로 갖는 이터러블이면서 동시에 이터레이터인 객체를 반환
 - Map.prototype.entries : 요소키, 요소값을 갖는 이터러블이면서 동시에 이터레이터인 객체를 반환